Journal of Computational and Applied Mathematics 236 (2012) 4318–4335

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Convex feasibility modeling and projection methods for sparse signal recovery

Avishy Carmi [a], Yair Censor [b], Pini Gurfil [a,*]

[a] Distributed Space Systems Lab, Faculty of Aerospace Engineering, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel

[b] Department of Mathematics, University of Haifa, Mt. Carmel, Haifa 31905, Israel

ARTICLE INFO

Article history:
Received 9 January 2011
Received in revised form 16 January 2012

Keywords:
Convex feasibility problems
Subgradient projection methods
Compressed sensing
Signal processing

ABSTRACT

A computationally-efficient method for recovering sparse signals from a series of noisy observations, known as the problem of compressed sensing (CS), is presented. The theory of CS usually leads to a constrained convex minimization problem. In this work, an alternative outlook is proposed. Instead of solving the CS problem as an optimization problem, it is suggested to transform the optimization problem into a convex feasibility problem (CFP), and solve it using feasibility-seeking sequential and simultaneous subgradient projection methods, which are iterative, fast, robust and convergent schemes for solving CFPs. As opposed to some of the commonly-used CS algorithms, such as Bayesian CS and Gradient Projections for sparse reconstruction, which become inefficient as the problem dimension and sparseness degree increase, the proposed methods exhibit robustness with respect to these parameters. Moreover, it is shown that the CFP-based projection methods are superior to some of the state-of-the-art methods in recovering the signal's support. Numerical experiments show that the CFP-based projection methods are viable for solving large-scale CS problems with compressible signals.

1. Introduction

Recent studies have shown that sparse signals can be recovered accurately using less observations than predicted by the Nyquist/Shannon sampling principle; the resulting theory is known as compressed sensing (CS) [1,2]. The essence of this new theory is a data acquisition formalism in which compression plays a fundamental role.

Sparse – and more generally – compressible signals arise naturally in many fields of science and engineering. Typical problems include the reconstruction of images from under-sampled Fourier data, biomedical imaging and astronomical observations [3,4]. Other applications include model-reduction methods enforcing sparseness for reducing computational complexity and storage capacities. The reader is referred to [1,2] for an extensive overview of the CS theory.

The issue of computational efficiency is important for large-scale CS problems. Ideally, a computational scheme for recovering sparse and/or compressible signals should be computationally efficient and accurately identify the signal support. Additional desirable features would be simple and transparent implementation (no "black boxes"), scalability, robustness with respect to some sparseness/compressibility measure, and ubiquitous applicability to real-world problems.

In this paper, we propose to look at the CS problem as a Convex Feasibility Problem (CFP) rather than a minimization problem. This different approach leads us to the use of different algorithms, namely, feasibility-seeking rather than minimization algorithms. CFPs are aimed at finding a point in the intersection set of a family of closed convex sets, and are not aimed at optimizing a performance index. The CFP is a fundamental modeling approach in numerous fields;

* Corresponding author.
E-mail addresses: avishy@aerodyne.technion.ac.il (A. Carmi), yair@math.haifa.ac.il (Y. Censor), pgurfil@aerodyne.technion.ac.il (P. Gurfil).

0377-0427/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2012.03.021

see, e.g., [5,6] and references therein. It has been used to model real-world problems in image reconstruction from projections [7], in radiation therapy treatment planning [8], and many more areas; see [9, Section 1].

Most often, CFPs are solved by performing projections onto the individual closed convex sets. This is carried out in various ways, by using different *projection methods*, resulting in a myriad of algorithms that usually exhibit good convergence while consuming low computational resources. Some of the projection algorithms may even provide predictable performance when the solution set of the CFP is empty [10]. In the language of [9]: "The main advantage of projection methods [for the CFP], which makes them successful in real-world applications, is computational. They commonly have the ability to handle huge-size problems of dimensions beyond which more sophisticated methods cease to be efficient or even applicable due to memory requirements. This is so because the building bricks of a projection algorithm are the projections onto the given individual sets, which are assumed to be easy to perform, and because the algorithmic structure is either sequential or simultaneous, or in-between, as in the block-iterative projection methods or in the more recently invented string-averaging projection methods". We refer the reader to [11] for a discussion on the connection between projection methods and variational inequalities; to [12,13] for some specific developments of projection methods; to [14, Chapter 5] for a textbook presentation; and to [15] for an extensive review.

However, projections onto convex sets may be difficult to perform when the individual constraints sets are nonlinear (i.e., not hyperplanes or half-spaces). This is so because computing orthogonal projections onto arbitrary convex sets requires a separate, inner-loop minimization for finding the distance minimizer between a given point and the set, a process that usually involves a considerable computational effort. An elegant alternative is to use *subgradient projections*. Subgradient projections only require the instantaneous subgradient in order to perform the next iteration. The projection is performed onto an intermediate point, and not directly onto the convex set. Some of the subgradient projection methods, such as Cyclic Subgradient Projections (CSP) [16] have a convergence proof when the intersection set of all the convex constraints is nonempty.

Here we use both CSP and Simultaneous Subgradient Projections (SSP) for efficiently solving the CS problem. The proposed algorithms are characterized by their (i) computational efficiency; (ii) accuracy in the sense of signal support recovery; (iii) robustness to varying compressibility and sparseness levels; and (iv) straightforward, transparent implementation. By using extensive numerical evaluations, we illustrate the advantages of the proposed scheme – in terms of computational efficiency and accuracy – compared to some other commonly-used methods for solving the CS problem.

2. Related work

The recovery of sparse signals consists of solving an NP-hard minimization problem [1,17]. State-of-the-art methods for addressing this optimization problem commonly utilize convex relaxations, non-convex local optimization and greedy search mechanisms. Convex relaxations are used in various methods such as Least Absolute Shrinkage and Selection Operator (LASSO) [18], Least Angle Regression (LARS) [19], the Dantzig Selector (DS) [20], Basis Pursuit (BP) and BP de-noising [21]. The LASSO and LARS schemes are essentially homotopy methods that exploit pivoting operations on submatrices from the complete sensing matrix (i.e., sub-sensing matrices) for yielding a solution path to the convex optimization problem. These methods turn out to be very efficient whenever the sparseness level is relatively high, owing to the fact that only a few submatrices need to be provided, corresponding to the instantaneous support of the underlying reconstructed signal. The DS and the BP variants, recast a linear program and employ either simplex or interior-point techniques for obtaining an optimal solution. Similarly to the homotopy-based approaches, these methods become computationally intensive when the number of elements in the support of the underlying signal increases.

Non-convex optimization approaches include Bayesian methodologies such as the Relevance Vector Machine (RVM), also known as Sparse Bayesian Learning [22], as well as stochastic search algorithms, which are mainly based on Markov Chain Monte Carlo (MCMC) techniques [23–25]. In virtue of their Bayesian mechanism and in contrast to other optimization approaches, these methods provide a complete statistical solution to the CS problem by means of a probability density function. Nevertheless, the intensive computational requirements of these methods render their applicability questionable in high-dimensional problems.

Recently, the Bayesian framework was utilized to create efficient CS schemes [26,27]. The Bayesian CS algorithm proposed in [26] exploits both a sparseness-promoting hierarchical prior and an RVM mechanism for deriving point estimates and statistical error bounds. This method was shown to outperform some of the commonly-used greedy schemes both in accuracy and speed. The work in [27] derived a pseudo-measurement-based Kalman filtering algorithm for recovering sparse signals from noisy observations in a sequential manner. This approach extends CS to accommodate stochastic linear filtering problems and, similarly to the aforementioned Bayesian methods, yields a complete statistical solution to the problem (a Gaussian distribution).

Notable greedy search algorithms are the Matching Pursuit (MP) [28], the Orthogonal MP (OMP) [29], and the Orthogonal Least Squares (OLS) [30]. Both MP and OMP minimize the reconstruction error by iteratively choosing elements from the sensing matrix. The OMP involves an additional orthogonalization stage and is known to outperform the conventional MP. The OLS works in a similar fashion, employing an orthogonal transformation of the original sensing matrix. The greedy algorithms are advantageous in terms of computational cost compared to other optimization schemes when the basis projections are sufficiently incoherent, and the number of elements in the support is relatively small. This, in turn, implies that their performance may deteriorate dramatically in common realistic settings wherein the underlying signals are compressible rather than sparse.

The scalability of a CS technique essentially refers to its viability for high-dimensional settings. This issue is the focus of [31], wherein a new Gradient Projection (GP)-based method is proposed for solving possibly large-scale CS problems. As pointed out by [31], large-scale methods require only matrix–vector products involving the sensing matrix. As demonstrated in [31], the GP algorithm is efficient in high-dimensional settings compared to the aforementioned methods. However, similarly to the homotopy methods, the practical implementation of the GP algorithm requires a delicate tuning of the l_1-norm bounding parameter, which greatly affects its convergence and optimality.

Finally, the theory of CS has drawn much attention to the convex relaxation methods. It has been shown that the convex l_1 relaxation yields an exact solution to the recovery problem provided that two conditions are met: (i) the signal is sufficiently sparse, and (ii) the sensing matrix obeys the *Restricted Isometry Property* (RIP) at a certain level. Another complementary result ensures high accuracy when dealing with noisy observations. Further analysis of this result subsequently yielded its probabilistic version, which is characterized by recovery 'with overwhelming probability' [1].

3. Sparse signal recovery and compressed sensing

3.1. Support identification

Formally, the CS problem can be divided into two main subproblems: (i) support identification; and (ii) recovery of entries in the identified support [2]. The relationship between these two subproblems is quite obvious. As long as the actual support is accurately identified, the lowest attainable reconstruction error would be reached. In practice, however, many CS methods fail to reach the lowest attainable reconstruction error even after correctly identifying the support of the underlying signal. Fortunately, this issue can be circumvented by employing a simple least squares (LS) scheme based on the obtained support [20]. If the support dimension is small compared to the number of observations (which is necessary for obtaining reasonable recovery errors [2]), the application of an additional LS scheme will only slightly increase the concomitant computational overhead.

The CFP-based methods suggested in this paper turn out to be highly efficient for identifying the support within a relatively small number of iterations, irrespectively of the problem dimension and sparseness degree. In some cases, however, the recovery accuracy may be inadequate, depending on tuning. However, as mentioned above, this problem can be alleviated by employing an LS scheme based on the identified support.

In many practical applications, the recovery accuracy may not always be very important; in these cases, sparse quantities facilitate selective procedures for moderating the computational burden, preventing overfitting, and reducing energy requirements. Some of these applications include sensor networks [32], multiagent and complex systems [33], random fields and Bayesian networks for pattern recognition [34], multiple object tracking and system identification [35], to name only a few. Typically, in such applications the exact magnitudes of the nonzero signal entries are not as important as identifying the support. Consequently, for applications in which the efficiency of support identification is a key issue, our CFP-based approaches can be competitive with – and even preferable to – some of the state-of-the-art CS methods.

Throughout this work, we choose to quantify the support identification performance of the examined CS methods by computing their associated LS recovery errors, that is, the errors obtained after applying an LS scheme based on the identified support (a procedure that essentially consists of thresholding the recovered entries. See Section 1.6 in [20] for further details).

3.2. Problem formulation

Consider a signal represented by a vector in the n-dimensional Euclidean space $\chi \in \mathbb{R}^n$, which is sparse in some domain, i.e., it can be represented using a relatively small number of projections in some known, possibly orthonormal, basis, $\psi \in \mathbb{R}^{n\times n}$. Thus, we may write

$$\chi = \psi x = \sum_{i=1}^{n} x_i \psi_i = \sum_{x_j \in \mathrm{supp}(x)} x_j \psi_j, \quad \|x\|_0 < n, \tag{1}$$

where $\mathrm{supp}(x)$ and $\|x\|_0$ (the zero norm) are the respective notations for the support of x and its dimension (i.e., the number of non-zero components of x), and ψ_i is the i-th column of the transpose ψ^T of the matrix ψ. The problem of compressed sensing considers the recovery of x (and therefore of χ) from a limited number, $m < n$, of incoherent and possibly noisy measurements (or, in other words, sensing a compressible signal from a limited number of incoherent measurements) [1]. The measurements/observations vector y obeys a linear relation of the form

$$y = H'\chi = Hx \tag{2}$$

where $H \in \mathbb{R}^{m\times n}$ and $H = H'\psi$. In many practical applications, the observation vector y may be either inaccurate or contaminated by noise. In this case, which will be referred to as the *stochastic CS problem*, an additional noise term is added to the right-hand side of (2).

In general, under certain limitations on the sparseness degree of x, denoted by

$$s := \|x\|_0, \tag{3}$$

an exact solution to the recovery problem can be obtained by solving a subset-selection problem of the form

$$\begin{cases} \min\limits_{x} \|x\|_0 \\ \text{subject to } \|y - Hx\|_2^2 \le \varepsilon \end{cases} \tag{4}$$

for a sufficiently small ε. However, problem (4) is known to be NP-hard, which implies that in practice, an optimizer cannot be computed efficiently.

In the late 1990's, the l_1-norm was suggested as a sparseness-promoting term in the seminal works that introduced the LASSO operator [18] and the BP [21]. Recasting the sparse recovery problem (4) using the l_1-norm provides a convex relaxation, making an efficient solution possible using a myriad of well-established optimization techniques. Commonly, there are two convex formulations that are proposed to replace (4): The quadratically-constrained linear program, which takes the form

$$\begin{cases} \min\limits_{x} \|x\|_1 \\ \text{subject to } \|y - Hx\|_2^2 \le \varepsilon \end{cases} \tag{5}$$

or the quadratic program

$$\begin{cases} \min\limits_{x} \|y - Hx\|_2^2 \\ \text{subject to } \|x\|_1 \le \varepsilon'. \end{cases} \tag{6}$$

Recently, it has been shown [1,2] that an accurate solution of (4) can almost always be obtained by solving the convex relaxation (5) provided that the sensing matrix H obeys the RIP. Roughly, the RIP implies that the columns of a given matrix nearly form an orthonormal basis. This property can be obtained by several random constructions, which guarantee the uniqueness of the sparse solution. In particular, an exact recovery is highly probable when using such matrices provided that

$$s = \mathcal{O}(m/\log(n/m)). \tag{7}$$

For an extensive overview of several RIP constructions and their role in CS, the reader is referred to [1,2].

In practice, the unknown signal $x = (x_i)_{i=1}^n$ may be *nearly* sparse, in the sense of having many relatively small components, which are not identically zero. Such representations, frequently encountered in real-world applications, are termed *compressible*. Most of the results in the CS literature naturally extend to the compressible case, assuming some behavior of the small nonzero components. Such a behavior is suggested in [2], where the compressible components sequence is assumed to decay according to the power law

$$|x_i| \le \kappa i^{(-1/r)}, \qquad |x_i| \ge |x_{i+1}|, \tag{8}$$

where $\kappa > 0$ and $r > 0$ are the radius of a weak l_r-ball to which x is confined, and a decay factor, respectively. In this case, a measure of the signal sparseness degree can be obtained as

$$\hat{s} = n - \text{card}\{i \mid 1 \le i \le n,\ |x_i| \le \varepsilon\} \tag{9}$$

for some sufficiently small $\varepsilon > 0$, where 'card' denotes the cardinality of a set.

4. Subgradient projections for compressed sensing

In this section, we outline the convex feasibility problem and two algorithms for a solution thereof: The cyclic subgradient projections (CSP) and the simultaneous subgradient projections (SSP). We explain how these algorithms and their variants, collectively referred to as *projection methods for the convex feasibility problems* [15], are implemented for efficiently solving the convex CS problem described by Eqs. (5) or (6).

4.1. The convex feasibility problem

Given p closed convex subsets $Q_1, Q_2, \ldots, Q_p \subseteq \mathbb{R}^n$ of the n-dimensional Euclidean space, expressed as

$$Q_i = \left\{ z \in \mathbb{R}^n \mid f_i(z) \le 0 \right\}, \tag{10}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function, the *convex feasibility problem* (CFP) is

$$\text{find a point } z^* \in Q := \cap_{i=1}^p Q_i, \tag{11}$$

consequently, it solves the system of convex inequalities

$$f_i(z) \le 0, \quad i = 1, 2, \ldots, p. \tag{12}$$

If $Q \neq \emptyset$ then the CFP is said to be *consistent*. The context of convex inequalities gives rise to the realm of subdifferential calculus, in which the definitions of subgradients and subdifferentials play a fundamental role. Given the convex function $f_i : \mathbb{R}^n \to \mathbb{R}$, a vector $t \in \mathbb{R}^n$ is called a *subgradient* of f_i at point $z^0 \in \mathbb{R}^n$ if

$$f_i(z) - f_i(z^0) \geq \langle t, z - z^0 \rangle, \quad \text{for all } z \in \mathbb{R}^n. \tag{13}$$

The subdifferential of f_i at z^0, denoted by $\partial f_i(z^0)$, is the nonempty compact convex set

$$\partial f_i(z^0) := \left\{ t \mid f_i(z) - f_i(z^0) \geq \langle t, z - z^0 \rangle, \ \forall z \in \mathbb{R}^n \right\}. \tag{14}$$

If f_i is differentiable at z^0, then the subgradient is unique and equal to the gradient $t = \nabla f_i(z^0)$.

4.2. Subgradient projections

Subgradient projections have been incorporated into iterative algorithms for the solution of CFPs; see, e.g., [15, Section 7]. The algorithms can be roughly divided into two main categories, *sequential* and *simultaneous*. The cyclic subgradient projections (CSP) method for the CFP is a sequential subgradient projections algorithm, see [16], and is summarized in Algorithm 1.

Algorithm 1 The method of cyclic subgradient projections (CSP)

Initialization: $z^0 \in \mathbb{R}^n$ is arbitrary.
Iterative step: Given z^k calculate the next iterate z^{k+1} by

$$z^{k+1} = \begin{cases} z^k - \alpha_k \dfrac{f_{i(k)}(z^k)}{\| t^k \|_2^2} t^k, & \text{if } f_{i(k)}(z^k) > 0, \\ z^k, & \text{if } f_{i(k)}(z^k) \leq 0, \end{cases} \tag{15}$$

where $t^k \in \partial f_{i(k)}(z^k)$ is a subgradient of $f_{i(k)}$ at the point z^k, and the relaxation parameters $\{\alpha_k\}_{k=0}^{\infty}$ are confined to an interval $\epsilon_1 \leq \alpha_k \leq 2 - \epsilon_2$, for all $k \geq 0$, with some, arbitrarily small, $\epsilon_1, \epsilon_2 > 0$.
Constraint-Index Control: The sequence $\{i(k)\}_{k=0}^{\infty}$ is cyclic, that is, $i(k) = k(\mathrm{mod} p) + 1$ for all $k \geq 0$, where p is the number of sets in the CFP.

A convergence result for the CSP method in the consistent case can be found in [14, Theorem 5.3.1], where it was shown that if the functions $f_i(z)$ are continuous and convex on $\mathbb{R}^n$ for all i; $Q := \cap_{i=1}^{p} Q_i \neq \emptyset$; and the subgradients are uniformly bounded, then any sequence $\{z^k\}_{k=0}^{\infty}$ produced by Algorithm 1 converges to a solution of the CFP, i.e., $\lim_{k\to\infty} z^k = z^*$. The convergence proof in [14, Chapter 5] is based on the concept of Fejér-monotonicity: A sequence $\{z^k\}_{k=0}^{\infty}$ is *Fejér monotone* with respect to some fixed set $Q \subseteq \mathbb{R}^n$, if for any $z \in Q$

$$\|z^{k+1} - z\|_2 \leq \|z^k - z\|_2, \quad \forall k \geq 0. \tag{16}$$

Sequential projection methods for solving CFPs usually have simultaneous counterparts. The simultaneous subgradient projections (SSP) method (see, e.g., [36,37]) is a simultaneous variant of the CSP, and is given in Algorithm 2. Convergence analyses for these algorithms are available for consistent ($Q \neq \emptyset$) CFPs.

Algorithm 2 The method of simultaneous subgradient projections (SSP)

Initialization: $z^0 \in \mathbb{R}^n$ is arbitrary.
Iterative step:

i. Given z^k calculate, for all $i \in \{1, 2, \ldots, p\}$, intermediate iterates $\zeta^{k+1,i}$ by

$$\zeta^{k+1,i} = \begin{cases} z^k - \alpha_k \dfrac{f_i(x^k)}{\| t^k \|^2} t^k, & \text{if } f_i(z^k) > 0, \\ z^k, & \text{if } f_i(z^k) \leq 0, \end{cases} \tag{17}$$

where $t^k \in \partial f_i(z^k)$ is a subgradient of f_i at the point z^k, and the relaxation parameters $\{\alpha_k\}_{k=0}^{\infty}$ are confined to an interval $\epsilon_1 \leq \alpha_k \leq 2 - \epsilon_2$, for all $k \geq 0$, with some, arbitrarily small, $\epsilon_1, \epsilon_2 > 0$.

ii. Calculate the next iterate z^{k+1} by

$$z^{k+1} = \sum_{i=1}^{p} w_i \zeta^{k+1,i} \tag{18}$$

where w_i are fixed, user-chosen, positive weights with $\sum_{i=1}^{p} w_i = 1$.

4.3. CSP and SSP for compressed sensing

Based on Eqs. (5) and (6), we reformulate the CS problem as follows: Given a real-valued $m \times n$ matrix H, a measurement vector $y \in \mathbb{R}^m$, a user-chosen $\varepsilon > 0$, and sparseness (or compressibility) information on the unknown vector $x \in \mathbb{R}^n$,

$$\begin{cases} \text{find } x^* \in \mathbb{R}^n \\ \text{such that} \\ y = Hx^* \\ \|x^*\|_1 \le \varepsilon. \end{cases} \tag{19}$$

This problem is not equivalent to the optimization formulations in (5) or (6). It proposes a different problem formulation for the CS problem that we further translate into the language of CFP. To do this, we define the m closed convex sets $Q_1, Q_2, \ldots, Q_m \subseteq \mathbb{R}^n$ by the hyperplanes

$$Q_i = \left\{ x \in \mathbb{R}^n \mid y_i = \langle h^i, x \rangle \right\}, \tag{20}$$

where $\langle \cdot, \cdot \rangle$ is the inner product and h^i is the i-th column of the matrix H^T. For these sets the subgradient is $t = \nabla(y_i - \langle h^i, x \rangle) = h^i$. An additional closed convex set $Q_{m+1} \subseteq \mathbb{R}^n$ is defined as

$$Q_{m+1} = \left\{ x \in \mathbb{R}^n \mid \|x\|_1 - \varepsilon \le 0 \right\}. \tag{21}$$

For $f_{m+1} = \|x\|_1 - \varepsilon$, using the notation of (10), we note that a subgradient (according to the definition of Eq. (13)) at the origin can be chosen so that

$$t \in \partial f_{m+1}(0), \quad t = \mathbf{1} \tag{22}$$

where $\mathbf{1}$ is the n-dimensional vector of ones. A subgradient t of $f_{m+1} = \|x\|_1 - \varepsilon$ at an arbitrary $x \in \mathbb{R}^n$ can be chosen so that

$$t = \operatorname{sign}(x) \quad \text{and} \quad \|t\|_2^2 = n, \tag{23}$$

where the vector $\operatorname{sign}(x)$ is defined by

$$\text{for all } 1 \le j \le n: \quad (\operatorname{sign}(x))_j = \operatorname{sign}(x_j) = \begin{cases} 1, & \text{if } x_j \ge 0, \\ -1, & \text{if } x_j < 0. \end{cases} \tag{24}$$

The next stage is to implement the CSP algorithm in order to seek a feasible solution $x^* \in Q := \cap_{i=1}^{m+1} Q_i$ for the CFP reformulation of the CS problem (19). Following the recipe of Algorithm 1, using the relationships (22)–(24), this is formulated in Algorithm 3. The iterations "pass" sequentially over the m sets Q_i followed by an iteration with respect to Q_{m+1} and then cyclically repeats the process. Note that ε in (26) and (28) is that of (19).

Algorithm 3 Cyclic subgradient projections for compressed sensing (CSP–CS)

Initialization: Select a starting point $x^0 \in \mathbb{R}^n$.
Iterative Step: For $k = 0, 1, 2, \ldots$, given the current iterate x^k, compute the next iterate x^{k+1} as follows:

1. Set $z^1 = x^k$
2. For $\ell = 1, 2, \ldots, m$, given z^ℓ calculate

$$z^{\ell+1} = z^\ell - \alpha_{\ell,k} \frac{\langle h^\ell, z^\ell \rangle - y_\ell}{\|h^\ell\|^2} h^\ell \tag{25}$$

3. From z^{m+1} compute the next iterate x^{k+1}

$$x^{k+1} = \begin{cases} z^{m+1} - \alpha_{m+1,k} \dfrac{\|z^{m+1}\|_1 - \varepsilon}{n} \operatorname{sign}(z^{m+1}), & \text{if } \|z^{m+1}\|_1 > \varepsilon, \\ z^{m+1}, & \text{if } \|z^{m+1}\|_1 \le \varepsilon. \end{cases} \tag{26}$$

and return to 1.

Relaxation parameters: The user-chosen relaxation parameters $\{\alpha_{\ell,k}\}_{\ell=1,\,k=0}^{m+1,\,\infty}$ are such that, for all ℓ and all k, $\epsilon_1 \le \alpha_{\ell,k} \le 2 - \epsilon_2$ for some arbitrary $\epsilon_1, \epsilon_2 > 0$.

An application of Algorithm 2 for the CS problem (19), termed SSP–CS, is done in a similar manner, yielding Algorithm 4.

Note that the inactive step of Algorithms 1 and 2, i.e., when the current iterate remains unchanged, is unnecessary if the constraints are linear equations (cf. [14, Chapter 6]). This fact helps to increase the computational efficiency. In the same context, it is worth noting that without using the 1-norm constraint implemented in Eq. (15), the recovery of x from the linear set of equations $y = Hx$ is possible, under certain regularity conditions, using the well-known algorithm of

Algorithm 4 Simultaneous subgradient projections for compressed sensing (SSP–CS)

Initialization: Select a starting point $x^0 \in \mathbb{R}^n$.
Iterative Step: For $k = 0, 1, 2, \ldots$, given the current iterate x^k, compute the next iterate x^{k+1} as follows:

1. For $\ell = 1, 2, \ldots, m$, calculate

$$z^{k,\ell} = x^k - \alpha_{\ell,k} \frac{\langle h^\ell, x^k \rangle - y_\ell}{\|h^\ell\|^2} h^\ell \tag{27}$$

2. Compute the vector $z^{k,m+1}$

$$z^{k,m+1} = \begin{cases} x^k - \alpha_{m+1,k} \dfrac{\|x^k\|_1 - \varepsilon}{n} \operatorname{sign}(x^k), & \text{if } \|x^k\|_1 > \varepsilon, \\ x^k, & \text{if } \|x^k\|_1 \le \varepsilon. \end{cases} \tag{28}$$

3. the next iterate x^{k+1} is the convex combination of the intermediate vectors $\{z^{k,\ell}\}_{\ell=1}^{m+1}$, i.e.,

$$x^{k+1} = \sum_{\ell=1}^{m+1} w_\ell z^{k,\ell} \tag{29}$$

where w_ℓ are fixed user-chosen positive weights with $\sum_{\ell=1}^{m+1} w_\ell = 1$.

Relaxation parameters: The user-chosen relaxation parameters $\{\alpha_{\ell,k}\}_{\ell=1,\,k=0}^{m+1,\,\infty}$ are such that, for all ℓ and all k, $\epsilon_1 \le \alpha_{\ell,k} \le 2 - \epsilon_2$ for some arbitrary $\epsilon_1, \epsilon_2 > 0$.

Kaczmarz [14, Chapter 6]. In this case, the obtained solution x^* would minimize $\|x\|_2$ if x^0 is chosen to be in the range of H^T. However, adding the 1-norm constraint is key to a successful recovery when x is sparse (or compressible).

The convergence of Algorithm 3 and its SSP variant, Algorithm 4, to some feasible solution x^* is guaranteed, provided that the CFP is consistent (i.e., the solution set is nonempty). However, for real-world large-scale problems, it is not useful to invest resources to determine *a priori* whether a given value of ε renders the CFP consistent. The behavior of the subgradient projections algorithms for the inconsistent case was investigated in [10], where it was shown that simultaneous subgradient projections methods usually behave better in the inconsistent case. However, simultaneous projection methods tend to converge slower than sequential methods; it is thus a good idea to combine these two methods in order to fight instabilities and obtain fast convergence. This issue is discussed in the next section.

4.4. Convergence analysis

As previously mentioned both methods, the CSP and the SSP, are convergent as long as the constraints are consistent (see, e.g., [36,37]). The proposed approach may, however, be inconsistent due to an unrefined setting of ε in either (26) or (28). In such a case Fejér monotonicity can be restored by properly choosing the relaxation parameters $\{\alpha_{m+1,k}\}_{k\ge 0}$. In particular, we consider the following cases.

4.4.1. Random ensemble sensing matrices

We propose a sensible approach for computing an upper bound on $\alpha_{m+1,k}$. For example, whenever the sensing matrix satisfies the property (called *low distortion embedding*)

$$\left| \|Hx\|_2^2 - 1 \right| \le \delta \tag{30}$$

for every unit-norm and s-sparse $x \in \mathbb{R}^n$ (i.e., x contains not more than s significant entries), and some $\delta \in (0, 1)$. Under some restrictions on the underlying dimensions, many random matrices are most likely to obey (30). These account for the Gaussian and Bernoulli ensembles, which are frequently encountered in the CS literature [38,39]. In general, however, similar properties can be obtained for any random ensemble matrix irrespective of the distribution of its entries [40].

Consider an ensemble sensing matrix $H^T \in \mathbb{R}^{n\times m}$ for which the columns h^i, $i = 1, 2, \ldots, m$ are realizations of a random vector $h \in \mathbb{R}^n$. For such a matrix the following theorem provides a computable upper bound on $\alpha_{m+1,k}$.

Theorem 1 (*Deterministic Version*). *Let H^T be a random ensemble matrix for which the columns are sampled from an arbitrary zero-mean distribution with a covariance $\frac{1}{m} I_{n\times n}$. Then, setting*

$$\alpha_{m+1,k} \le 2\left(1 - \frac{\sqrt{d}}{\sqrt{1-\delta}} \frac{E\left\{\|y\|_2^2\right\}^{1/2}}{\|x^k\|_1}\right) \tag{31}$$

where the right-hand side term in this inequality is positive, renders the output of either the CSP–CS or SSP–CS Fejér monotone (cf. (16)) with respect to the convex set defined by the problem's constraints. The expectation operator, denoted as $E\{\cdot\}$, is taken

with respect to the entries of H. In addition, $d \in [s, n]$, and $\delta \in (0, 1)$ is an isotropic coefficient pertaining to the matrix H. Particularly, for $s = \mathcal{O}(m/\log n)$

$$\delta = c\sqrt{\frac{\log n}{m}} E\left\{\|h\|_2^{\log m}\right\}^{1/\log m} < 1 \tag{32}$$

where $c > 0$.

In cases where the statistics of the measurements cannot be determined and hence $E\left\{\|y\|_2^2\right\}$ is unspecified in (31), an alternative – probabilistic – variant of Theorem 1 is given as follows.

Theorem 2 (*Probabilistic Version*)**.** *Consider a Gaussian ensemble sensing matrix or otherwise a sensing matrix whose (zero-mean) entries are uniformly distributed with a variance $\frac{1}{m}$. Let also*

$$\alpha_{m+1,k} \le 2\left(1 - \frac{\sqrt{d}}{\sqrt{1-\delta}} \frac{\|y\|_2}{\|x^k\|_1}\right) \tag{33}$$

where the right-hand side term in this inequality is assumed to be positive. If in addition $s = \mathcal{O}(m/\log n)$ then with probability exceeding $1 - \mathcal{O}(e^{-\gamma n})$, $\gamma > 0$, the output of either the CSP–CS or SSP–CS is Fejér monotone (cf. (16)) with respect to the convex set defined by the problem's constraints.

The proofs of both theorems are provided in the Appendix. Lastly, we note that reasonable values for δ are below 0.5, where for Gaussian ensembles $\delta \approx 0.5$ (see for example [38,39]).

4.5. Practical implementation

A few issues concerning our practical implementation of the CSP–CS and SSP–CS algorithms are elaborated below.

1. *Stopping Criteria*: The CSP–CS and SSP–CS routines are terminated when either some predetermined maximal number of iterations is exceeded, or when there is no significant difference between two consecutive estimates, viz. $\|x^{k+1} - x^k\|_2 \le \gamma$ for some small $\gamma > 0$.
2. *Refinements*: As mentioned in the previous section, in some cases the family of constraints may not be consistent due to the setting of ε, which in turn implies that the convergence of the CSP method is not guaranteed. Typically, in such scenarios the estimated parameters persistently fluctuate around some nominal value. The reconstructed signal itself can be highly accurate; however, its entries do not attain a fixed value. This problem can be alleviated by heuristically incorporating an additional refinement stage, according to one of the following methods:
   (a) *Gauss–CSP (LS-augmented CSP)*: After a given number of CSP–CS iterations, the support of the unknown signal can be readily approximated based on the magnitudes of the estimated elements. As soon as the support is given, the signal can be accurately reconstructed following a simple LS procedure. This approach is inspired by the strategy adopted in [20] for correcting the bias inherent to the Dantzig Selector (the corrected scheme was referred to in [20] as the Gauss–Dantzig Selector).
   (b) *A slightly different implementation of the Gauss–CSP*: Referred to as *alternating Gauss-CSP*; this implementation often dramatically improves the convergence properties of the CSP method. It consists of incorporating the LS stage directly into the CSP algorithm in an alternating manner. This is done by interlacing the LS routine whenever the iteration index k in Algorithm 3 reaches an arbitrary predetermined value. In our experience, this variant of the Gauss–CSP typically converged much faster than any other alternative.
   (c) *CSP–SSP*: Normally, running a few SSP iterations subsequently to the termination of the CSP routine improves the accuracy of recovery. This stems from the improved behavior of the SSP method in the inconsistent case, discussed previously.
3. *Block-iterative projections*: In some programming environments, it might be more computationally efficient to process a group of q observations

   $$y_i = \langle h^i, x\rangle, \quad i = j,\ j+1, \ldots, j+q \tag{34}$$

   at each iteration rather than a single one (as in CSP) or all (as in SSP). Such an approach was termed in the literature of projection methods for CFPs Block-Iterative Projections (BIP), see, e.g., [41,15]. Our own implementation of the CSP–CS exploits this idea for alleviating the workload on the MATLAB ® interpreter, which may become prohibitively slow when loops are involved. This approach does not require any considerable modification of the original algorithm. In practice, this is accomplished by running the cyclic index from 1 to $p/q + 1$ while the CSP–CS update (25) is repeated for q observations at each iteration.

5. Illustrative examples

In this section, our newly-proposed algorithms are assessed based on extensive comparisons with some of the commonly-used methods for CS. The algorithms considered herein consist of the homotopy method LARS [19], the two greedy

algorithms, OMP [29] and BP [21], the Gradient Projection (GP)-based method of [31], and the Bayesian CS (BCS) of [26]. The Web links for the MATLAB ® implementations of these methods are provided in the Appendix.

5.1. Performance measures

In order to highlight the weaknesses and virtues of the various methods, we examine here both synthetic and realistic scenarios, associated with the two signal types frequently encountered in CS applications: Sparse and compressible (nearly sparse in the sense of (9)). An indicator for the difficulty of recovery is assigned to each problem based on its unique settings (dimension and sparseness degree). This measure, termed here *the recovery index*, is derived from the relation [1] $m \geq c(s \log n)$ (for some $c > 0$), and is given by

$$\rho := (s/m) \log n. \tag{35}$$

This index essentially refers to the probability of recovery assuming the sensing matrix obeys the RIP up to a certain sparseness degree. As was already pointed out in [1], as this measure increases, an exact recovery becomes less probable.

Throughout this section, the signal reconstruction error e is computed as

$$e := \sqrt{\frac{\|x - x^*\|_2^2}{d(x)}} \tag{36}$$

where x^* is the reconstructed signal and the normalizing term $d(x)$ is determined according to the signal type,

$$d(x) := \begin{cases} \sum_{i=1}^{n} \min(x_i^2, \sigma^2), & \text{for sparse } x, \\ \|x\|_2^2, & \text{for compressible } x, \end{cases} \tag{37}$$

with σ being the standard deviation of the observation noise. The formulation (37) corresponds to both the *ideal* (sparse x) and the *normalized* (compressible x) recovery measures that are used for assessing the reconstruction accuracy in [20,26], respectively. The mean reconstruction error $\bar{e}$ is computed in a similar fashion over N runs as

$$\bar{e} := \sqrt{N^{-1} \sum_{i=1}^{N} e^2(i)}, \tag{38}$$

where $e(i)$ denotes the error in the i-th run.

5.2. Sparse signal recovery

In the first scenario, the various methods are applied for the recovery of a sparse signal from noisy observations.

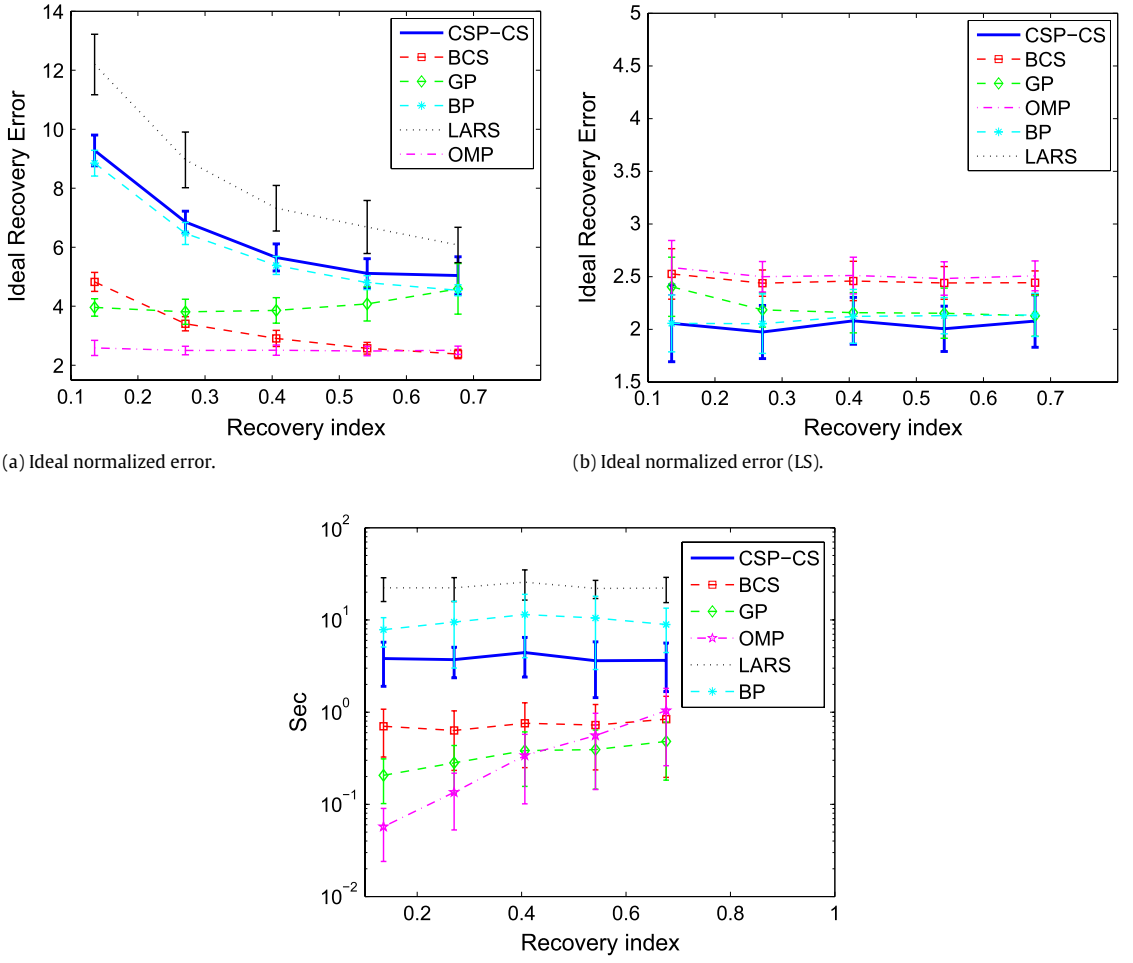
The sensing matrix $H \in \mathbb{R}^{m \times n}$ used here consists of normalized random entries sampled from a zero-mean Gaussian distribution with a standard deviation of $1/\sqrt{m}$. This type of random construction has been shown to obey the RIP up to a reasonable sparseness level [2]. We examine the recovery performance of the CS algorithms in various settings consisting of different problem dimensions, ranging from 512×1024 through 1024×2048 and 2048×4096 to 3072×6144, and different sparseness levels. The original signal x is composed of only few nonzero random components, which are uniformly sampled over $[-1, 1]$, and of which the indices are randomly picked between 1 and n. In all runs, the measurement noise standard deviation is set to 0.01. The various algorithms' tuning parameters are taken as those that minimize the recovery error based on tuning runs. The CSP–CS parameters are set to

$$\alpha_{\ell,k} = 1.8, \quad \text{for } \ell = 1, 2, \ldots, m, \qquad \varepsilon = 10^{-4}, \tag{39}$$

$$\text{and} \quad \frac{\alpha_{m+1,k}}{n} = \begin{cases} 70^{-2}, & \text{If } k \leq 2000, \\ 100^{-2}(1 + k/10^4)^{-1}, & \text{Otherwise}, \end{cases} \tag{40}$$

which yielded the best accuracy and convergence time. The CSP routine is terminated when either the number of iterations exceeds 5000 or when the normed difference between two consecutive estimates satisfies $\|x^{k+1} - x^k\|_2 < \gamma$, where $\gamma = 0.01$ for the low-dimensional problems and $\gamma = 0.5$ otherwise.

The averaged performance over 50 Monte Carlo runs of the various methods for the problem dimension of 512×1024 is depicted in Fig. 1 for different recovery indices (cf. (35)) ranging from $\rho = 0.1$ to $\rho = 0.7$. This figure shows both the mean ideal recovery error (cf. (37) and (38)) and the corresponding mean convergence time along with their standard deviations, which are illustrated using error bars. Observing the top left panel in this figure reveals a performance hierarchy that places the CSP–CS as the 2nd worst right after the LARS and just a bit before the BP. The remaining methods attain higher accuracy in this case.



(a) Ideal normalized error.

(b) Ideal normalized error (LS).

(c) Convergence time.

Fig. 1. The average performance (over 50 Monte Carlo runs) of the CS methods (including the computationally intensive LARS and BP) for problem dimension 512×1024. The performance in the sense of support identification is demonstrated using the LS-augmented variants in the upper right figure.

Nevertheless, the story completely changes when examining the LS-augmented methods in the top right panel. Here the CSP–CS attains the *best accuracy* over the entire range of sparseness degrees. Following the argument in Section 3.1, this detail readily implies that the CSP–CS exhibits improved support identification capabilities compared to the other CS methods.

By observing the bottom panel of Fig. 1, it can be easily recognized that the CSP–CS is the 3rd slowest method in this case. The other computationally intensive methods here are the LARS and the BP. In this example, the convergence times of all methods excluding those of the greedy OMP and the GP roughly stays unchanged over the entire range of sparseness levels. The extreme slant of the OMP line is due to the nature of this algorithm, which tends to become computationally intensive as the assumed maximal support size increases.

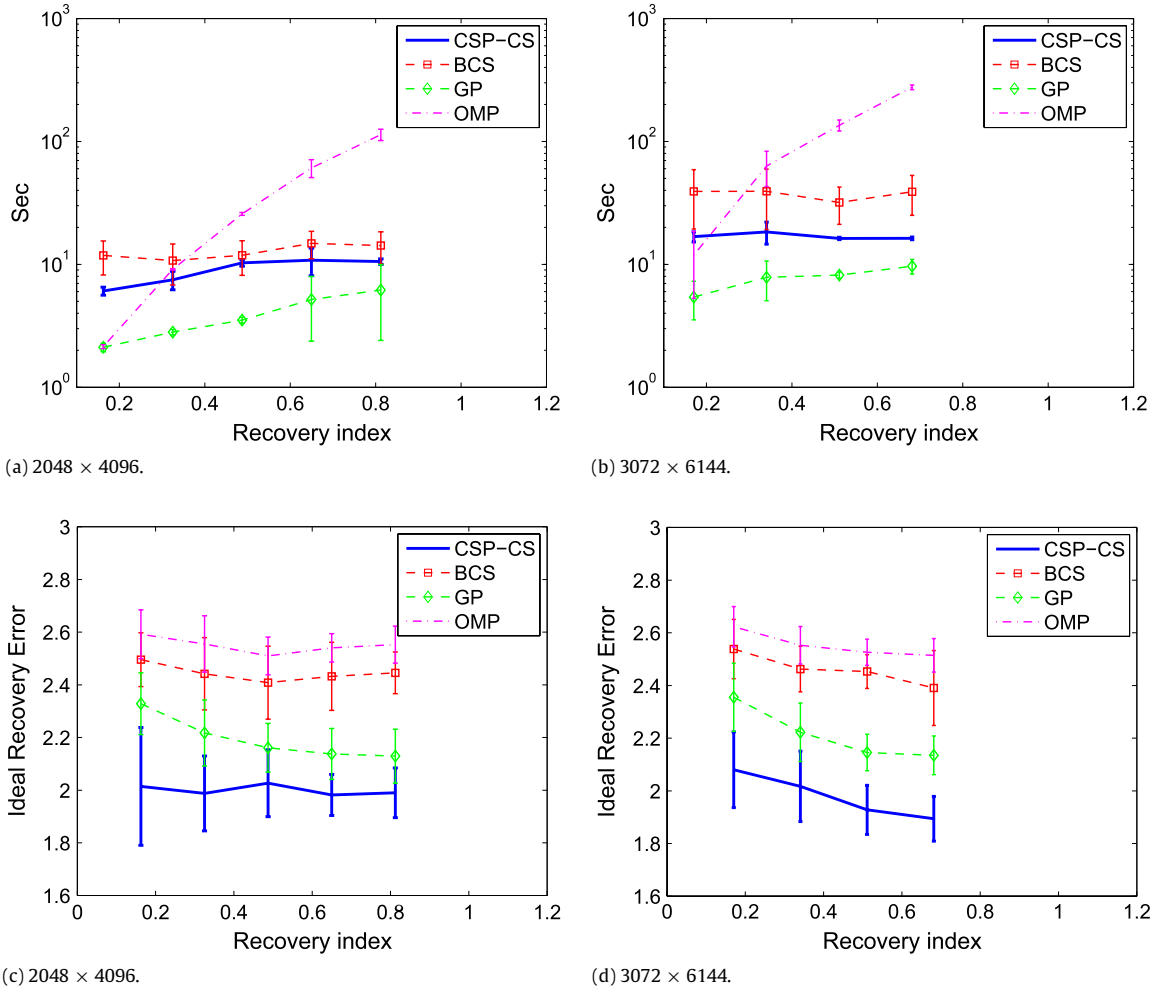
The above insights are further emphasized in Table 1, which focuses on the numerical values obtained in these experiments for two nominal recovery indices, $\rho = 0.1$ and $\rho = 0.6$. The boldface values in this table correspond to the averaged recovery errors of the LS-augmented methods. Thus, it can be clearly seen that the CSP–CS attains the best accuracy in the sense of support identification, yielding a mean associated recovery error of around 2.05.

The performance of the algorithms for the problem sizes of 2048×4096 and 3072×6144 is illustrated in Fig. 2. Here we have excluded the computationally intensive methods, LARS and BP, as their convergence time became prohibitively long. The performance of the remaining methods for the various problem dimensions is illustrated via two types of panels. The upper panel shows the convergence times for different sparseness levels, whereas the bottom panel shows the corresponding ideal recovery errors averaged over 50 Monte Carlo runs. As before, the standard deviations from the mean values are depicted using error bars.

By observing the upper panel in Fig. 2, we see that the CSP–CS method is faster than the BCS over the entire range of sparseness levels for both problem dimensions. A similar conclusion applies when comparing the CSP–CS with the OMP

Table 1
Recovery of sparse signals. The ideal recovery error (left columns) and convergence time (right columns) of the various methods for the problem dimension 512 × 1024. The bold values correspond to the accuracy of the LS-augmented variants. Averaged over 50 Monte Carlo runs. In this example the CSP–CS exhibits the best support identification capabilities as manifested by its associated (bold) recovery errors.

Method	Recovery errors				Convergence times	
	$\rho = 0.1$		$\rho = 0.6$		$\rho = 0.1$ (s)	$\rho = 0.6$ (s)
LARS	12.19	**12.19**	6.07	**6.07**	22.28	22.18
BP	8.84	**2.05**	4.54	**2.13**	7.84	8.93
OMP	2.56	**2.58**	2.50	**2.50**	0.05	1.03
BCS	4.82	**2.52**	2.37	**2.44**	0.70	0.83
GP	3.95	**2.40**	4.58	**2.12**	0.20	0.48
CSP–CS	9.28	**2.05**	5.04	**2.07**	3.81	3.64



(a) 2048 × 4096.

(b) 3072 × 6144.

(c) 2048 × 4096.

(d) 3072 × 6144.

Fig. 2. Attainable recovery errors after support identification. The average performance of the CS algorithms for two problem dimensions and sparseness degrees. Showing the convergence time (upper panels) and the ideal recovery error (bottom panels). The computationally excessive methods (LARS and BP) are not shown here.

from a certain sparseness degree corresponding to a recovery index of around 0.4. The CSP–CS turns out to be *significantly faster* than both the BCS and the OMP over almost the entire range of sparseness levels for the larger problem dimension.

The associated recovery errors of the various methods are presented in the bottom panels in Fig. 2. From these panels we notice that, as before, the CSP–CS exhibits the best support recovery performance among the examined CS methods.

The above insights are supported by Table 2, which provides the timing and estimation error values for two nominal recovery indices. It shows that the CSP–CS algorithm outperforms the other methods in terms of support identification accuracy, as manifested by its associated recovery error, and is also the 2nd fastest method (after the GP). Another interesting

Table 2
Attainable recovery errors after support identification. The ideal recovery error (left columns) and convergence time (right columns) of the various methods for the problem dimension 3072×6144. Shown results are averaged over 50 Monte Carlo runs.

Method	Recovery errors		Convergence times	
	$\rho = 0.1$	$\rho = 0.6$	$\rho = 0.1$ (s)	$\rho = 0.6$(s)
OMP	2.62	2.52	11.89	135.52
BCS	2.53	2.45	39.18	31.84
GP	2.35	2.14	5.41	8.18
CSP–CS	2.08	1.92	16.79	16.26

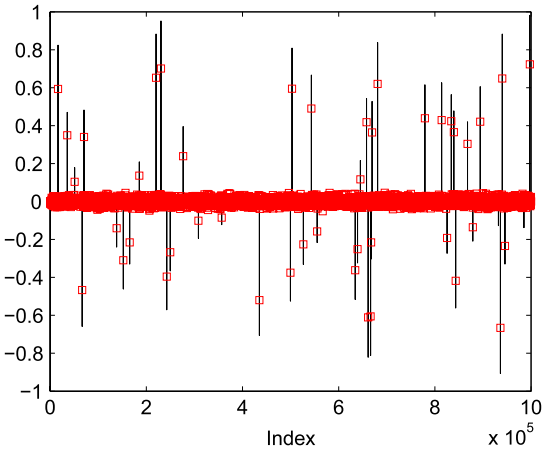


Fig. 3. Support identification performance after 160 iterations for a problem dimension of 3000×10^6.

and important detail that comes about from both Table 2 and Fig. 2 is related to the fact that, as opposed to the GP the running time of which is highly sensitive to the sparseness degree, the CSP–CS computation time remains almost unchanged with respect to this factor. This observation could, in fact, be expected, as the CSP–CS mechanism does not really distinguish between the elements in the support and the elements that are not. This feature renders the CSP–CS highly robust and efficient when applied either to compressible problems or in such scenarios where the recovery index is relatively large.

The support recovery performance of the CSP–CS after 160 iterations in a large-scale scenario is illustrated in Fig. 3. In this example, the problem dimension is 3000×10^6 and the recovery index is 0.23 (i.e., a support size of 50). This figure shows that the CSP–CS manages to identify 48 out of 50 of the most prominent non-zero entries.

A comparison of the various CSP implementations that were discussed in Section 4.5 is provided in Fig. 4. This figure demonstrates the convergence properties of the methods based on a single run for a problem dimension of 2048×4096. Thus, it can be easily recognized that the plain CSP–CS attains the worst recovery error, as it begins to fluctuate around some nominal value. As pointed out previously, this behavior indicates that the family of constraints in this case is inconsistent, owing to an improper setting of the parameter ε. This problem is alleviated in both of the variants, the CSP–SSP and the Gauss–CSP. Although both these methods outperform the plain CSP–CS, it seems that the best attainable error is achieved by using the Gauss–CSP, whereas the fastest convergence is obtained by using the alternating CSP–LS scheme.

5.3. Compressible examples

Compressible signals are of greater practical importance when it comes to real-world applications. As mentioned earlier, such signals are nearly sparse in the sense that they consist of many relatively small elements that are not identically zero. Therefore, we consider here two realistic experimental studies involving compressible signals. The first example, which is conducted in the spirit of the previous synthetic one, consists of constructing the Discrete Fourier Transform (DFT) of an undersampled time series. The second experiment involves the reconstruction of the Shepp–Logan head phantom used for assessing the performance of recovery schemes in tomography.

5.3.1. Constructing a DFT from undersampled data

In this example we consider a discrete signal y in the time domain, which takes the form

$$y_k = \sum_{i=1}^{n_f} \sin(\omega_i t_k), \quad k = 1, 2, \ldots, n \tag{41}$$

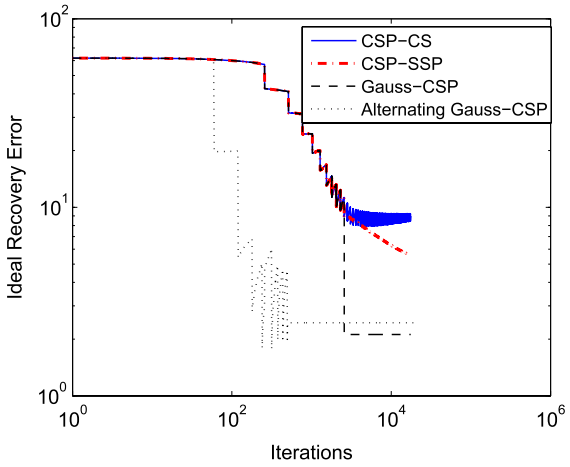


Fig. 4. The ideal recovery error of the CSP variants. Single run, problem dimension 2048 × 4096.

where the frequencies $\omega_i,\ i = 1, 2, \ldots, n_f$ are uniformly sampled over the interval $[\pi, 10\pi]$. Let $x = (x_k)_{k=1}^n \in \mathbb{R}^n$ be the DFT of y, that is

$$x_k = (1/\sqrt{n}) \sum_{j=1}^{n} y_j \exp(-2\pi(j-1)(k-1)i/n) \tag{42}$$

which can be written compactly as $x = Fy$ with F being the unitary DFT matrix. Now, suppose that we wish to reconstruct x from a randomly sampled vector $(y_j)_{j=1}^m$ for which $m \ll n$. In other words, we attempt to reconstruct the DFT x from an undersampled time series. This is a classical CS problem, for which the sensing matrix H is given as a partial inverse DFT matrix consisting of only m randomly-picked rows. Thus,

$$y = H_m x \tag{43}$$

where H_m is composed of m randomly picked rows from F^* (where the asterisk denotes the conjugate transpose), corresponding to the components of y. The vector x itself is expected to be compressible, essentially comprised of decaying coefficients in the vicinity of those that are associated with the underlying frequencies $\omega_i,\ i = 1, 2, \ldots, n_f$.

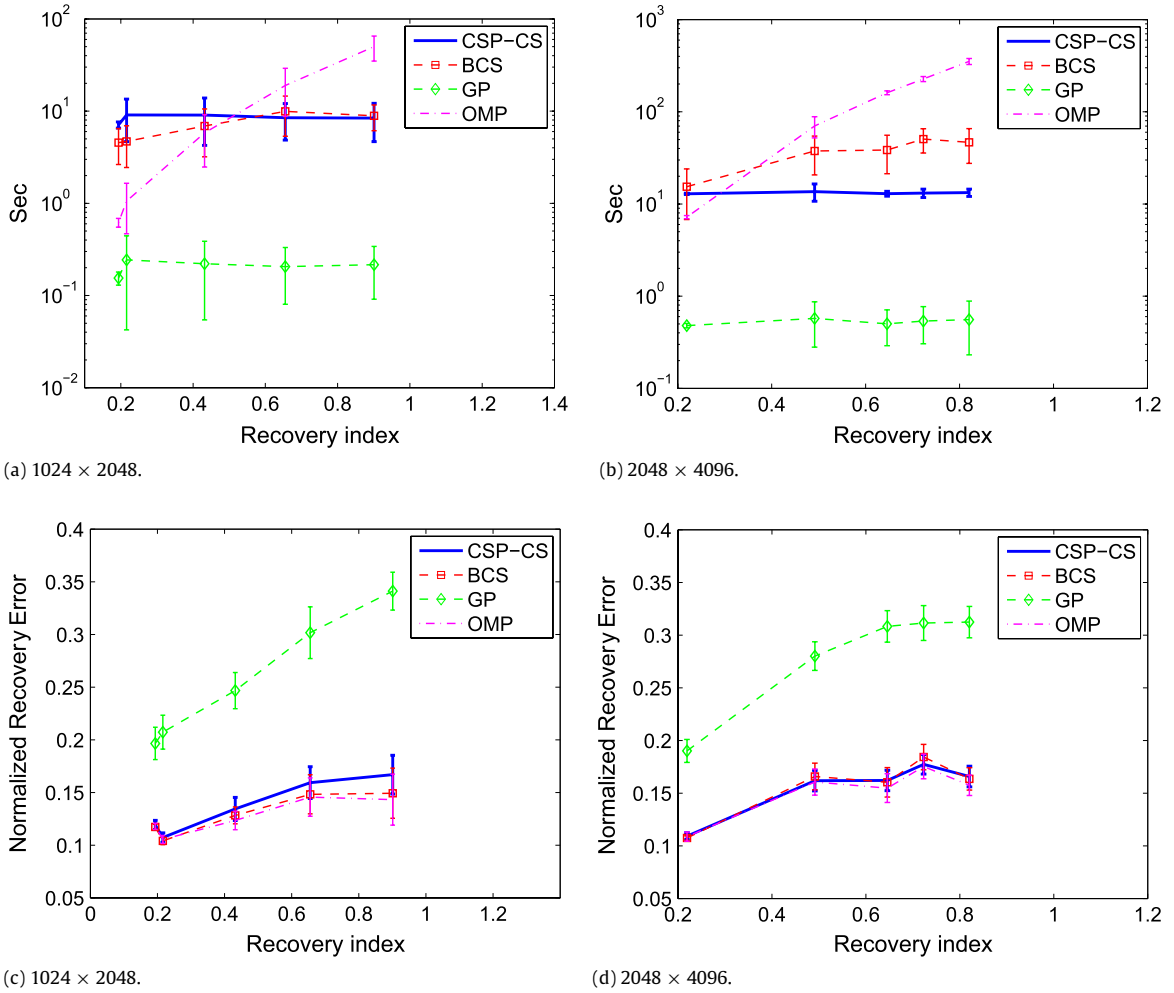
Similarly to the synthetic case, we apply the various *non augmented* CS methods (excluding the computationally excessive ones) for different problem dimensions and sparseness levels. As distinct from the previous example, here the recovery index (35) is computed based on the *effective* sparseness measure (9) with $\varepsilon = 0.05 \max_i |x_i|$ (here, the maximum is taken with respect to the actual signal entries, so the threshold value, used for determining the cardinality of the support, is set to 5% of the largest peak). The CS algorithms' tuning parameters are chosen to maximize accuracy based on tuning runs. The CSP relaxation parameters and termination conditions remain unchanged.

The averaged performance of the various methods over 50 Monte Carlo runs (in which a new set of frequencies $(w_i)_{i=1}^{n_f}$ is sampled at the beginning of each run) is depicted in Fig. 5. The upper panel in this figure shows the CS methods' mean convergence times and their associated standard deviations (error bars) for different problem dimensions (ranging from 512 × 1024 through 1024 × 2048 to 2048 × 4096) and effective sparseness levels (corresponding to recovery indices of between 0.1 and 0.8). The corresponding recovery errors of the various methods are provided in the bottom panel of this figure.

By observing both these panels, we see that the CSP–CS method maintains the *best tradeoff* between accuracy and computational load as the problem becomes more complex, as indicated by both its dimensionality and sparseness degree. In virtue of its underlying mechanism, the CSP–CS exhibits robustness with respect to both these factors as it attains recovery errors that are comparable to those of the BCS and the OMP at a nearly fixed computational cost. This renders it the fastest reliable method among the methods tested here as the problem dimension increases. Although the GP is the fastest scheme in this case, its accuracy is extremely low with recovery errors of nearly the magnitude of the signal itself. The LS-augmented GP does possess a clear advantage over the unaugmented one; however, its performance is still unsatisfactory (with recovery errors of almost twice than those of the other methods).

The above insights are further illustrated in Table 3, in which the timing and recovery error values are repeated for the problem dimension 2048 × 4096. It can be seen from this table that the CSP–CS attains a recovery error that is similar to those of the BCS and the OMP. The advantage of the CSP–CS over the other methods becomes prominent as the recovery index increases. Thus, the CSP–CS maintains a nearly fixed convergence time over the entire range of recovery indices, essentially yielding the best accuracy and timing tradeoff among the methods tested here for the extremal index value of 0.8.

Fig. 6 depicts the performance of the CSP–CS in recovering typical sparse and compressible signals of nearly the same recovery index. This figure suggests that, in practice, though the recovery indices are nearly the same, it might be



(a) 1024 × 2048.

(b) 2048 × 4096.

(c) 1024 × 2048.

(d) 2048 × 4096.

Fig. 5. Recovering a DFT from undersampled data. Showing the average performance of the CS algorithms for two problem dimensions and effective sparseness degrees. The convergence time and the normalized error are depicted in the upper and lower panels, respectively.

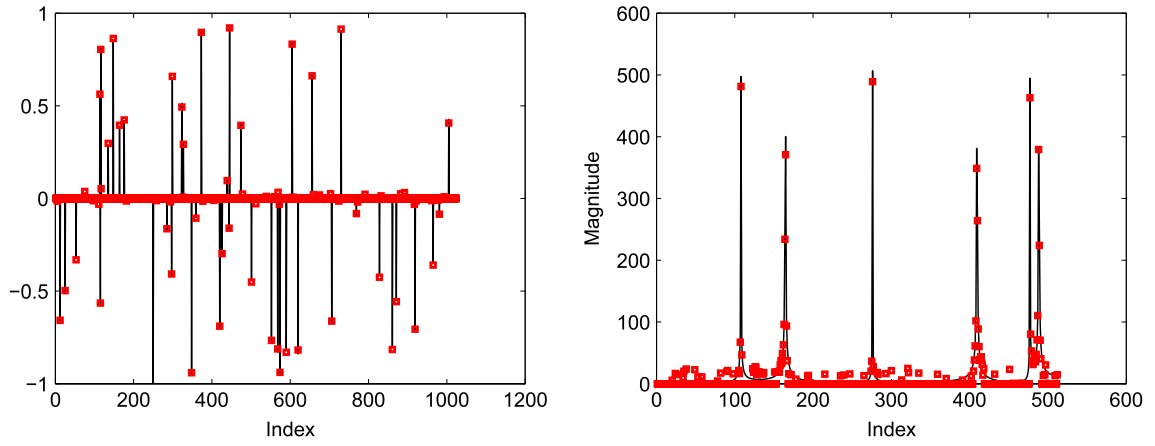
Table 3
Recovering a DFT from undersampled data. The normalized recovery errors (left columns) and convergence times (right columns) of the various methods for the problem dimension 2048 × 4096. Shown results are averaged over 50 Monte Carlo runs.

Method	Normalized recovery errors		Convergence times	
	$\rho = 0.2$	$\rho = 0.8$	$\rho = 0.2$ (s)	$\rho = 0.8$(s)
OMP	0.10	0.17	7.20	226.80
BCS	0.10	0.18	15.40	50.62
GP	0.19	0.31	0.47	0.53
CSP–CS	0.10	0.17	12.87	13.11

more difficult to reconstruct a compressible representation rather than a sparse one. This follows from the fact that the compressible estimate on the right panel is less accurate than its companion on the left.

5.3.2. Image recovery from undersampled radial Fourier coefficients

In the last part of this section, we demonstrate the performance of some of the methods in recovering an image using undersampled 2D Fourier coefficients that are computed along radial lines. The example considered here follows along the lines of [20], where the Shepp–Logan phantom head image is used. In our experiment, however, we use a low-dimensional 128 × 128 version of this image, which yields a signal of dimension 128 × 128 = 16 384. We examine the CS methods for two scenarios in which the 2D-Fourier coefficients are sampled along either 32 or 64 radial lines (corresponding to nearly 25% or 50% of the available data).



(a) Sparse signal. (b) DFT (compressible) signal.

Fig. 6. Illustration of the recovery performance of the CSP–CS method for sparse and compressible signals.

(a) CSP–CS, $e = 0.12$. (b) GP, $e = 0.19$. (c) Original.

Fig. 7. The original and reconstructed Shepp–Logan phantom head. Showing the CSP–CS and the GP methods. The recovery errors are based on 64 projections, which is equivalent to 50% undersampled data.

(a) CSP–CS, $e = 0.22$. (b) GP, $e = 0.28$. (c) Original.

Fig. 8. The original and reconstructed Shepp–Logan phantom head. Showing the CSP–CS and the GP methods. The recovery errors are based on 32 projections, which is equivalent to 25% undersampled data.

Three methods are applied for recovering the phantom head image: BCS, GP and CSP–CS (unaugmented versions). However, the performance of only two of them, the GP and the CSP–CS are shown in Figs. 7 and 8 as the BCS exhibited poor performance, yielding recovery errors of nearly the signal magnitude (specifically, around 0.9). Both these figures clearly show the superiority of the CSP–CS over the GP in both scenarios.

6. Conclusions

A new approach to compressed sensing (CS) was introduced. The new method utilized the instantaneous subgradient for projecting the previous iterate on a convex set, thereby approaching a feasible point of the underlying convex feasibility problem resulting from a family of possibly nonlinear, convex constraint sets.

The Cyclic Subgradient Projection (CSP) method, constituting the heart of this approach, facilitates the efficient solution of large-scale CS problems since it involves vector inner products only. An extensive numerical comparison of our CSP–CS algorithm and its variants with some of the state-of-the-art CS schemes demonstrates its advantages in high-dimensional compressible settings.

In particular, the CFP-based methods introduced here have been experimentally shown to be efficient for identifying the support of a sparse signal. Since support identification is a main difficulty in recovering sparse signals, the superior performance of the CSP–CS algorithm in support identification constitutes a convenient stepping stone for the possible subsequent application of refinement mechanisms, such as those based on least squares.

The proposed methods maintain a nearly fixed computational cost irrespective of the problem dimension and sparseness level; moreover, they easily cope with realistic scenarios involving high-dimensional compressible signals. Finally, extensive numerical comparisons show that the CSP–CS maintains the best tradeoff between computational efficiency and accuracy as the problem dimension and sparseness level increase.

Acknowledgments

The work of Y. Censor is supported by Grant No. 2009012 from the United States–Israel Binational Science Foundation (BSF) and by US Department of Army award number W81XWH-10-1-0170.

Appendix

In order to prove the theorems we will need the following lemma.

Lemma 1. *Let x^k be either one of the CSP–CS or SSP–CS iterates preceding the CS stages (26) or (28), respectively. Assume that $\|x^k\|_1 > \|x^*\|_1$ where x^* is the optimizer of the CS problem (19). Then for*

$$0 < \alpha_{m+1,k} \leq 2\left(1 - \frac{\|x^*\|_1}{\|x^k\|_1}\right) < 2 \tag{44}$$

we have that

$$\|x^{k+1} - x^*\|_2^2 \leq \|x^k - x^*\|_2^2, \quad \forall k \geq 0, \tag{45}$$

where x^{k+1} denotes the iterate following the CS stage ((26) or (28)).

Proof. Let

$$\lambda_k = \alpha_{m+1,k} \frac{g(x^k)}{\|t^k\|_2^2}, \tag{46}$$

where $g(x^k) = \|x^k\|_1 - \varepsilon$, and t^k denotes a subgradient of g computed at x^k. Following an argument similar to that in [14, Theorem 5.3.1] we may write

$$\|x^{k+1} - x^*\|_2^2 = \|x^k - \lambda_k t^k - x^*\|_2^2 = \|x^k - x^*\|_2^2 + \lambda_k^2 \|t^k\|_2^2 + 2\lambda_k \langle t^k, x^* - x^k \rangle. \tag{47}$$

Recalling that the subgradient t^k satisfies

$$\langle t^k, x^* - x^k \rangle \leq g(x^*) - g(x^k) = \|x^*\|_1 - \|x^k\|_1, \tag{48}$$

(47) yields

$$\|x^{k+1} - x^*\|_2^2 \leq \|x^k - x^*\|_2^2 + \lambda_k^2 \|t^k\|_2^2 + 2\lambda_k (\|x^*\|_1 - \|x^k\|_1). \tag{49}$$

Further, substituting (46) into (49) and rearranging, we obtain

$$\|x^{k+1} - x^*\|_2^2 \leq \|x^k - x^*\|_2^2 + \left(\alpha_{m+1,k}^2 - 2\alpha_{m+1,k} + 2\alpha_{m+1,k} \frac{\|x^*\|_1}{\|x^k\|_1}\right) \frac{g^2(x^k)}{\|t^k\|_2^2}. \tag{50}$$

Since $\frac{\|x^*\|_1}{\|x^k\|_1} \leq 1$, (45) readily follows. □

Note that the bound on $\alpha_{m+1,k}$ in (44) becomes tighter as the optimal solution x^* is approached which essentially entails slower and refined convergence assuming $\alpha_{m+1,k}$ is properly regulated. Yet, since the optimizer is unknown this bound does not really convey useful information about how large $\alpha_{m+1,k}$ can be in practice. This premise can be changed however by considering Theorems 1 and 2 whose proofs are given next.

A.1. Proof of Theorem 1

The underlying ensemble matrix approximates a convex body in a nearly isotropic position [40]. Using the key theorem from [40] we observe that

$$E\left\{\left|\|Hx\|_2^2-\|x\|_2^2\right|\right\}=E\left\{\left|x^T(H^TH-I_{n\times n})x\right|\right\}\le E\left\{\left\|(H^TH-I_{n\times n})\right\|_\sigma\right\}\|x\|_2^2$$
$$\le \underbrace{c\frac{\sqrt{\log n}}{\sqrt{m}}E\left\{\|h\|_2^{\log m}\right\}^{1/\log m}}_{\delta}\|x\|_2^2 \tag{51}$$

where $\|\cdot\|_\sigma$ denotes the matrix spectral norm. As pointed out in [40], this inequality holds for $\delta<1$ which entails $m\ge cn\log n$. A much better lower bound on m is obtained upon recalling that x is s-sparse. Thus, following a rationale similar to that in [40] it can be readily verified that (51) holds for $m\ge cs\log n$, or otherwise $s=\mathcal{O}(m/\log n)$.

Invoking Jensen's inequality, (51) yields

$$\left|E\left\{\|Hx\|_2^2\right\}-\|x\|_2^2\right|\le E\left\{\left|\|Hx\|_2^2-\|x\|_2^2\right|\right\}\le\delta\|x\|_2^2 \tag{52}$$

implying

$$(1-\delta)\|x^*\|_2^2\le E\left\{\|y\|_2^2\right\}\le(1+\delta)\|x^*\|_2^2 \tag{53}$$

owing to the fact that $y=Hx^*$ where x^* is the optimizer of the CS problem. Hence,

$$\|x^*\|_1\le\sqrt{d}\|x^*\|_2\le\frac{\sqrt{d}}{\sqrt{1-\delta}}E\left\{\|y\|_2^2\right\}^{1/2} \tag{54}$$

where the left-hand side inequality is due to the equivalence of l_1 and l_2 norms. The parameter d therefore indicates the support cardinality of x^* which can be between s and n. Finally, substituting (54) into (44), Lemma 1 yields the theorem. □

A.2. Proof of Theorem 2

Under the conditions of the theorem, and in particular for $s=\mathcal{O}(m/\log n)$, the considered sensing matrix H obeys the restricted isometry property [38]

$$\left|\|Hx\|_2^2-\|x\|_2^2\right|\le\delta\|x\|_2^2 \tag{55}$$

for every s-sparse $x\in\mathbb{R}^n$ with probability of at least $1-\mathcal{O}(e^{-\gamma n})$ (for the actual values of γ see [2]). Recalling that $y=Hx^*$ where x^* is the unknown optimizer of the problem, (55) implies

$$\|x^*\|_1\le\sqrt{d}\|x^*\|_2\le\frac{\sqrt{d}}{\sqrt{1-\delta}}\|y\|_2 \tag{56}$$

where the left-hand side inequality is due to the equivalence of l_1 and l_2 norms. The parameter d therefore indicates the support cardinality of x^* which can be between s and n. Finally, substituting (56) into (44), Lemma 1 yields the theorem. □

A.3. Web locations of relevant CS algorithms

The MATLAB ® files implementing our new algorithms are available at http://www.technion.ac.il/pgurfil/csp-cs/. All the other source files for the various CS algorithms implemented herein can be found at the following locations:

Least Angle Regression (LARS): http://www.mathworks.com/matlabcentral/fileexchange/23186-lars-algorithm.
Basis Pursuit (BP): http://www.acm.caltech.edu/l1magic/.
Bayesian CS (BCS): http://people.ee.duke.edu/lihan/cs/.
Gradient Projection for Sparse Reconstruction (GPSR): http://www.lx.it.pt/mtf/GPSR/.

References

[1] E.J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Transactions on Information Theory 52 (2006) 489–509.
[2] E.J. Candès, Compressive sampling, in: Proceedings of the International Congress of Mathematicians, European Mathematical Society, Madrid, Spain, 2006, pp. 1433–1452.
[3] M. Lustig, D. Donoho, J.M. Pauly, Sparse MRI: the application of compressed sensing for rapid MR imaging, Magnetic Resonance in Medicine 58 (2007) 1182–1195.
[4] U. Gamper, P. Boesiger, S. Kozerke, Compressed sensing in dynamic MRI, Magnetic Resonance in Medicine 59 (2008) 365–373.
[5] P.L. Combettes, The foundations of set-theoretic estimation, Proceedings of the IEEE 81 (1993) 182–208.
[6] P.L. Combettes, The convex feasibility problem in image recovery, Advances in Imaging and Electron Physics 95 (1996) 155–270.
[7] G.T. Herman, Fundamentals of Computerized Tomography: Image Reconstruction from Projections, scoend ed., Springer, NY, 2009.
[8] Y. Censor, M.D. Altschuler, W.D. Powlis, On the use of Cimmino's simultaneous projections method for computing a solution of the inverse problem in radiation therapy treatment planning, Inverse Problems 4 (1988) 607–623.
[9] Y. Censor, W. Chen, P.L. Combettes, R. Davidi, G.T. Herman, On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints, Computational Optimization and Applications 51 (2012) 1065–1088.
[10] D. Butnariu, Y. Censor, P. Gurfil, E. Hadar, On the behavior of subgradient projections methods for convex feasibility problems in Euclidean spaces, SIAM Journal on Optimization 19 (2008) 786–807.
[11] I. Yamada, Hybrid steepest descent method for variational inequality problem over the fixed point set of certain quasi-nonexpansive mappings, Numerical Functional Analysis and Optimization 25 (2004) 619–655.
[12] G. Crombez, Non-monotoneous parallel iteration for solving convex feasibility problems, Kybernetika 39 (2003) 547–560.
[13] G. Crombez, A sequential iteration algorithm with non-monotoneous behaviour in the method of projections onto convex sets, Czechoslovak Mathematical Journal 56 (2006) 491–506.
[14] Y. Censor, S.A. Zenios, Parallel Optimization: Theory, Algorithms, and Applications, Oxford University Press, New York, NY, USA, 1997.
[15] H.H. Bauschke, J.M. Borwein, On projection algorithms for solving convex feasibility problems, SIAM Review 38 (1996) 367–426.
[16] Y. Censor, A. Lent, Cyclic subgradient projections, Mathematical Programming 24 (1982) 233–235.
[17] R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization, IEEE Signal Processing Letters 14 (2007) 707–710.
[18] R. Tibshirani, Regression shrinkage and selection via the LASSO, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1996) 267–288.
[19] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Annals of Statistics 32 (2004) 407–499.
[20] E. Candès, T. Tao, The dantzig selector: statistical estimation when p is much larger than n, Annals of Statistics 35 (2007) 2313–2351.
[21] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Journal of Scientific Computing 20 (1998) 33–61.
[22] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, Journal of Machine Learning Research 1 (2001) 211–244.
[23] R.E. McCulloch, E.I. George, Approaches for Bayesian variable selection, Statistica Sinica 7 (1997) 339–374.
[24] B.A. Olshausen, K.J. Millman, Learning sparse codes with a mixture-of-Gaussians prior, in: Advances in Neural Information Processing Systems (NIPS), 2000, pp. 841–847.
[25] P.J. Wolfe, S.J. Godsil, W. Ng, Bayesian variable selection and regularization for time–frequency surface estimation, Journal of the Royal Statistical Society 66 (2004) 575–589.
[26] S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing, IEEE Transactions on Signal Processing 56 (2008) 2346–2356.
[27] A. Carmi, P. Gurfil, D. Kanevsky, Methods for sparse signal recovery using Kalman filtering with embedded pseudo-measurement norms and quasi-norms, IEEE Transactions on Signal Processing 58 (2010) 2405–2409.
[28] S. Mallat, Z. Zhang, Matching pursuits with time–frequency dictionaries, IEEE Transactions on Signal Processing 4 (1993) 3397–3415.
[29] Y.C. Pati, R. Rezaiifar, P.S. Krishnaprasad, Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, in: Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers, IEEE, 2002, pp. 40–44.
[30] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their application to non-linear system identification, International Journal of Control 50 (1989) 1873–1896.
[31] M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, IEEE Journal on Selected Topics in Signal Processing 1 (2007) 586–597.
[32] S. Joshi, S. Boyd, Sensor selection via convex optimization, IEEE Transactions on Signal Processing 57 (2009) 451–462.
[33] S.I. Roumeliotis, G.A. Bekey, Distributed multi-robot localization, IEEE Transactions on Robotics and Automation 18 (2002) 781–795.
[34] J.E. Taylor, K.J. Worsley, Detecting sparse signals in random fields, with an application to brain mapping, Journal of the American Statistical Association 102 (2007) 913–928.
[35] E. Wang, J. Silva, L. Carin, Compressive particle filtering for target tracking, in: IEEE/SP 15th Workshop on Statistical Signal Processing, IEEE, 2009, pp. 233–236.
[36] A.N. Iusem, L. Moledo, A finitely convergent method of simultaneous subgradient projections for the convex feasibility problem, Computational and Applied Mathematics 5 (1986) 169–184.
[37] L.T. Dos Santos, A parallel subgradient method for the convex feasibility problem, Journal of Computational and Applied Mathematics 18 (1987) 307–320.
[38] E.J. Candes, The restricted isometry property and its implications for compressed sensing, Comptes Rendus Mathematique 346 (2008) 589–592.
[39] R.G. Baraniuk, M.A. Davenport, D. Ronald, M.B. Wakin, A simple proof of the restricted isometry property for random matrices, Constructive Approximation 28 (2008) 253–263.
[40] M. Rudelson, Random vectors in the isotropic position, Journal of Functional Analysis 164 (1999) 60–72.
[41] R. Aharoni, Y. Censor, Block-iterative projection methods for parallel computation of solutions to convex feasibility problems, Linear Algebra and its Applications 120 (1989) 165–175.